



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/727,183	12/03/2003	Miroslav Cina	19797-0011001/2003P00384U	5201

32864 7590 12/24/2008
FISH & RICHARDSON, P.C.
PO BOX 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER

TRUONG, CAM Y T

ART UNIT	PAPER NUMBER
----------	--------------

2169

NOTIFICATION DATE	DELIVERY MODE
-------------------	---------------

12/24/2008

ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

PATDOCTC@fr.com

**UNITED STATES DEPARTMENT OF COMMERCE****U.S. Patent and Trademark Office**

Address : COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

APPLICATION NO./ CONTROL NO.	FILING DATE	FIRST NAMED INVENTOR / PATENT IN REEXAMINATION	ATTORNEY DOCKET NO.
---------------------------------	-------------	---	---------------------

10727183

12/3/2003

CINA, MIROSLAV

19797-

0011001/2003P00384U

EXAMINER

Cam Y T. Truong

FISH & RICHARDSON, P.C.
PO BOX 1022
MINNEAPOLIS, MN 55440-1022

ART UNIT**PAPER**

2169

20081219

DATE MAILED:

Please find below and/or attached an Office communication concerning this application or proceeding.

Commissioner for Patents

On 12/15/2008, Examiner faxed a proposed amendment to Attorney Thomas H. Reger II. Examiner has been waiting the applicant's response to the proposed amendment. Applicant has one month to response to this communication. The amendment as indicated below:

1. (Currently amended) A method of scheduling access to a database by multiple processes, comprising:
in multiple instances of a software-implemented procedure, each instance of the procedure associated with a particular one of the multiple processes associating a lock level with a particular process and attempting to associate a different lock level with the particular process until the process is granted access to the database, a higher lock level representing a larger number of other processes having priority over the particular process in accessing the database, each of the processes being associated with no more than one lock level;
repeatedly calling, using the particular process, the associated instance of the procedure to attempt to associate the particular process with successively lower lock levels until the lock level is equal to a preset value, and each time the particular process has been successfully associated with a lower lock level, releasing a previous lock level associated with the particular process so that the previous lock level is available to be associated with other processes, the preset value is equal to one;
allowing, using the software-implemented procedure, the particular process to access the database when the lock level for the particular process is equal to a preset value,
updating, using the software-implemented procedure, data indicating which lock level is associated with which process;
wherein the particular process changes two or more lock levels from an initially assigned lock level to the lock level having the preset value before the process is allowed to access the record;
storing in a queue information indicating which process is associated with which lock level;
updating the information indicating which lock level is associated with which process;
attempting to access the record, by a process, earlier than another process is assigned a lower lock level than the other process, and each process other than the process terminating its access to the record is assigned a lower lock level when the process terminates its access to the record, each of the processes attempts to associate itself with a lower lock level independently of other processes;
locking the record when the lock level having the preset value is associated with a process, wherein at least two of the processes are being run in a parallel processing environment;
allowing processes to read the record but not modify the record when the lock levels for the processes are different from the preset value.

Claim 26 will be amended as claim 1.

9. (Currently Amended) A method of scheduling access to a database by multiple processes comprising:
in multiple instances of a software-implemented procedure,
upon receiving a request from a first process to access a record in a database,
associating a first lock level with the first process and allowing the first process to access the record, preventing other processes from modifying the record until the first process finishes accessing the record, a higher lock level representing a larger number of other processes having priority over the particular process in accessing the database, each of the processes being associated with no more than one lock level;
at a computer, upon receiving a request from a second process to access the record while the first process is still accessing the record, associating a second lock level with the second process, the computer includes a processor;
upon receiving a request from a third process to access the record while the process is still accessing the record, associating a third lock level with the third process;
when the first process finishes accessing the record, releasing the first lock level; the second and third processes each repeatedly calling one of the instances of the procedure respectively associated with the second or third process to attempt to associate itself with a lower lock level until the lock level is equal to a preset value, the preset value is equal to one;
when one of the second and third processes associates itself with the first lock level, permitting the process to modify the record; and
updating, using the instances of the software-implemented procedure data indicating which lock level is associated with which process;
wherein if the second process associates itself with the first lock level before the third process associates itself with the first lock level, the third process associates itself with the second lock level and repeatedly attempts to associate itself with the first lock level;
reassigning the lock levels of the processes when a process accessing the record terminates its access to the record;
storing in a queue information indicating which process is associated with which lock level;
updating the information indicating which lock level is associated with which process;
attempting to access the record, by a process, earlier than another process is assigned a lower lock level than the other process, and each process other than the process terminating its access to the record is assigned a lower lock level when the process terminates its access to the record, each of the processes attempts to associate itself with a lower lock level independently of other processes;
locking the record when the lock level having the preset value is associated with a process, wherein at least two of the processes are being run in a parallel processing environment;
allowing processes to read the record but not modify the record when the lock levels for the processes are different from the preset value.

claim 14 will be canceled.

Claim 34 will be amended as claim 9.

19. (Currently Amended) A system comprising:
a database to store records;
a queue to store information relating to lock levels of processes that attempt to access the records, each different process having a different lock level when attempting to access the same record, one of the processes having a particular lock level, the process having the particular lock level being allowed to access the record, a higher lock level representing a larger number of other processes having priority over the particular process in accessing the database, each of the processes being associated with no more than one lock level; and
a programmable processor to:
execute multiple instances of a procedure, each instance of the procedure associated with one of the processes, to assign a respective lock level to each of the different processes, each of at least some of the different processes having a lock level other than the particular lock level repeatedly calling one of the instances of the procedure associated with it to attempt to associate itself with another lock level that is closer to the particular lock level until the process associates itself with the particular lock level until the another lock level is equal to a preset value until the lock level is equal to a preset value, the procedure allowing any one process to access the record when that one process has the particular lock level;
wherein each of some of the processes changes two or more lock levels from an initially assigned lock level to the particular lock level having the preset value before the process is allowed to access the record;
the preset value is equal to one;
storing in a queue information indicating which process is associated with which lock level;
updating the information indicating which lock level is associated with which process;
attempting to access the record, by a process, earlier than another process is assigned a lower lock level than the other process, and each process other than the process terminating its access to the record is assigned a lower lock level when the process terminates its access to the record, each of the processes attempts to associate itself with a lower lock level independently of other processes;
locking the record when the lock level having the preset value is associated with a process, wherein at least two of the processes are

being run in a parallel processing environment.

.

/Cam Y Truong/
Primary Examiner, Art Unit 2169